

Exceções

- Um erro para qual pode existir um tratamento.

Exemplo

```
int x = 0;  
try {System.out.println(x/0);}   
catch (ArithmeticException e)  
    {System.out.println("Exemplo de  
tratamento de exceção!");}
```

Exceções comando `throw` e cláusula `throws`

- Quando um método pode causar um exceção o mesmo deve possuir uma cláusula **`throws`**.
- Para lançar um exceção usamos o comando **`throw`**.

Exemplo...

ExemploExcecao.java

Exercício

- Escreva um programa simples para ler os valores de um dia, um mês e um ano e criar um objeto da classe *Data*. O programa deve exibir uma janela de diálogo contendo a data representada pelo objeto criado, caso essa data seja válida, ou exibir uma janela contendo uma mensagem de erro, caso a data seja inválida. Você irá usar exceções para controlar possíveis erros na formação da data.

Finally

- Colocado após um bloco `catch`

Sempre é executado!!!

Estrutura

```
try{...}
```

```
catch(){...}
```

```
finally{...}
```

Exemplo

- Exemplo_finally.java
- ExemploRethrow.java- ilustra a propagação de exceções.

Quando usar **Finally**?

- Quando for tentar alocar um recurso, como por exemplo, um arquivo. Você pode então liberá-lo.

Exemplo

```
try
{
    /* recursos alocados */
    /* recursos usados */
}
catch(){...}
catch(){...}
finally
{...
    /*liberação dos recursos (com ou sem exceções)*/
}
```